What is claimed is:

1. A method of reducing a size of data difference representations, the method comprising:

identifying an original version of an input data stream in an original form;

dividing the original form of the original version of the input data stream into one or more separate original version output data streams through the use of a pre-processor;

identifying an updated version of the input data stream in the original form;

dividing the original form of the updated version of the input data stream into one or more separate updated version output data streams through the use of the pre-processor; and

differencing each of the one or more separate updated version output data streams with a corresponding original version output data stream to produce data difference representations.

2. The method of claim 1, wherein the data difference representations are smaller than a data difference representation created by differencing the original form of the updated version of the input data stream with the original form of the original input data stream.

3. The method of claim 1, further comprising:

reconstructing the one or more separate updated version output data streams from the data difference representations and the original version output data streams; and

combining the one or more separate updated version output data streams into the original form of the updated version of the input data stream through the use of a post-processor.

4. The method of claim 1 wherein the original form of the original version of the input data stream is empty.

5. The method of claim 1 wherein the pre-processor comprises decompression algorithms.

6. The method of claim 1 wherein the dividing steps separate volatile components of the input data stream from less volatile components of the input data stream.

7. The method of claim 6 wherein the input data stream is executable code.

8. The method of claim 7 wherein the volatile components comprise branch targets.

9. The method of claim 7 wherein the volatile components comprise data addresses.

10. The method of claim 7 wherein the less volatile components comprise instruction code.

11. The method of claim 7 wherein the less volatile components comprise immediate data.

12. The method of claim 1, further comprising:
packaging the data difference representations into a single data stream;
compressing the single data stream; and
storing the single data stream.

13. The method of claim 12, further comprising:
transmitting the single data stream;
uncompressing the single data stream; and
unpackaging the single data stream into the data difference representations.

14. A system of reducing a size of data difference representations, the system comprising:

a network;

a first computer system coupled to the network;

a system memory coupled to the first computer system, wherein the system memory stores one or more computer programs executable by the first computer system;

wherein the computer programs are executable to:

identify an original version of an input data stream in an original form;

divide the original form of the original version of the input data stream into one or more separate original version output data streams through the use of a pre-processor;

identify an updated version of the input data stream in the original form;

divide the original form of the updated version of the input data stream into one or more separate updated version output data streams through the use of the pre-processor; and

difference each of the one or more separate updated version output data streams with a corresponding original version output data stream to produce data difference representations.

15. The system of claim 14, wherein the data difference representations are smaller than a data difference representation created by differencing the original form of the updated version of the input data stream with the original form of the original input data stream.

16. The system of claim 14, further comprising:

a second computer system coupled to the network;

a system memory coupled to the second computer system, wherein the system memory stores one or more computer programs executable by the second computer system;

wherein the pre-processor is located in the first computer system; and

wherein the post-processor is located in the second computer system.

17. The system of claim 16, wherein the computer programs are further executable to:

reconstruct the one or more separate updated version output data streams from the data difference representations and the original version output data streams; and

combine the one or more separate updated version output data streams into the original form of the updated version of the input data stream through the use of a post-processor.

18. The system of claim 16 wherein the original form of the original version of the input data stream is empty.

19. The system of claim 16 wherein the pre-processor comprises decompression algorithms.

20. The system of claim 16 wherein the dividing steps separate volatile components of the input data stream from less volatile components of the input data stream.

21. The system of claim 16 wherein the input data stream is executable code.

22. The system of claim 21 wherein the volatile components comprise branch targets.

23. The system of claim 21 wherein the volatile components comprise data addresses.

24. The system of claim 21 wherein the less volatile components comprise instruction code.

25. The system of claim 21 wherein the less volatile components comprise immediate data.

26. The system of claim 16, wherein the computer programs are further executable to:

package the data difference representations into a single data stream;

compress the single data stream; and

store the single data stream on a memory medium coupled to the first computer system.

27. The system of claim 26, wherein the computer programs are further executable to:

transmit the single data stream from the memory medium coupled to the first computer system to the second computer system over the network;

uncompress the single data stream; and

unpackage the single data stream into the data difference representations.

28. A carrier medium which stores program instructions, wherein the program instructions are executable to implement reducing a size of data difference representations comprising:

identifying an original version of an input data stream in an original form;

dividing the original form of the original version of the input data stream into one or more separate original version output data streams through the use of a pre-processor;

identifying an updated version of the input data stream in the original form;

dividing the original form of the updated version of the input data stream into one or more separate updated version output data streams through the use of the pre-processor; and

differencing each of the one or more separate updated version output data streams with a corresponding original version output data stream to produce data difference representations.

29. The carrier medium of claim 28, wherein the data difference representations are smaller than a data difference representation created by differencing the original form of the updated version of the input data stream with the original form of the original input data stream.

30. The carrier medium of claim 28, wherein the program instructions are further executable to implement:

reconstructing the one or more separate updated version output data streams from the data difference representations and the original version output data streams; and

combining the one or more separate updated version output data streams into the original form of the updated version of the input data stream through the use of a post-processor.

31. The carrier medium of claim 28 wherein the original form of the original version of the input data stream is empty.

32. The carrier medium of claim 28 wherein the pre-processor comprises decompression algorithms.

33. The carrier medium of claim 28 wherein the dividing steps separate volatile components of the input data stream from less volatile components of the input data stream.

34. The carrier medium of claim 33 wherein the input data stream is executable code.

35. The carrier medium of claim 34 wherein the volatile components comprise branch targets.

36. The carrier medium of claim 34 wherein the volatile components comprise data addresses.

37. The carrier medium of claim 34 wherein the less volatile components comprise instruction code.

38. The carrier medium of claim 34 wherein the less volatile components comprise immediate data.

39. The carrier medium of claim 28, wherein the program instructions are further executable to implement:

packaging the data difference representations into a single data stream;

compressing the single data stream; and

storing the single data stream on a memory medium coupled to a first computer system.

40. The carrier medium of claim 39, wherein the program instructions are further executable to implement:

transmitting the single data stream from the memory medium coupled to the first computer system to a second computer system over a computer system network;

uncompressing the single data stream; and

unpackaging the single data stream into the data difference representations.

41. The carrier medium of claim 28, wherein the carrier medium is a memory medium.